IMPACT OF PARALLEL COMPUTING ON HIGH FIDELITY BASED MULTIDISCIPLINARY ANALYSIS

Guru P. Guruswamy
NASA Ames Research Center, Moffett Field, CA 94035-1000
Ph (650) 604 6329, Fax : (415) 650 1095, email: gguruswamy@mail.arc.nasa.gov

Abstract

An efficient super modular process to simulate aeroelasticity of aerospace vehicles using high fidelity flow equations such as the Euler/Navier-Stokes equations is presented. The process is suitable for both tightly coupled and uncoupled analysis. The process is designed to execute on massively parallel processors (MPP) and work-station clusters based on a multiple-instruction, multiple-data (MIMD) architecture. The fluids discipline is parallelized using a zonal approach whereas the structures discipline is parallelized using the sub-structures concept. Provision is also made to include controls domain. Computations of each discipline are spread across processors using IEEE standard message passing interface (MPI) for inter processor communications. Disciplines can run in parallel using a macro utility MPIRUN developed based on MPI. In addition to discipline parallelization and coarse-grain parallelization of the disciplines, embarrassingly parallel capability to run multiple parameter cases is implemented using a script system. The combined effect of three levels of parallelization is an almost linear scalability for multiple concurrent analyses that perform efficiently on MPP.

Introduction

Modern design requirements for aerospace vehicles push current technologies used in the design process to their limits or sometimes require more advanced technologies to meet the requirements. One of the many essential things needed to improve the performance is accurate prediction of aeroelastic characteristics using high fidelity methods. Though significant progress has taken place in high fidelity single discipline codes such as NASTRAN (ref. 1) for structures and OVERFLOW (ref. 2) for fluids, the effort to combine these single discipline codes into a multidiscipline code or a process is still in progress. Several attempts are made to expand single discipline codes to multidiscipline codes for e.g. ENSAERO (ref. 3), ENS3ADE (ref. 4), STARS (ref. 5) etc. Major disadvantage of all these codes is that they are tightly dependent on pre-selected individual disciplines. Due to rapid progresses that may take place in individual disciplines, a freedom is needed to replace them with improved ones. This requires a different approach than traditional code development.

One of the major disadvantages of using codes with high fidelity methods is the need for excessive requirements of computer resources, both in memory and speed. The start of the High Performance Computing and Communication Project (HPCCP) of NASA (ref. 6) initiated new ways of solving individual disciplines with scalable performance on multiple processors. Use of the IEEE standard Message Passing Interface (MPI) (ref. 7) utility led to successful parallel solution procedure. In order to couple different discipline domains communication between domains is accomplished through an interface at the end of each time step. For aeroelastic computations that involves fluids and structural domains, the aerodynamic loads are converted into the structural loads through the fluid-structural interface. Furthermore, the structural deformation is passed to the fluid domain through the interface. Then, the surface grid is deformed according to the structural deformation. In addition, control surface deflection computed in controls domain is superimposed on the deformed surface grid.

The overall communication design is shown in Fig. 1. In using the MPI library, a communicator is used to identify a group of processors so that a processor can communicate with others within the same group. Each group is represented by a box defined by dashed lines as shown in Fig. 1. In this case, however, only one processor is assigned to each group for a single coupled analysis. All the allocated processors have a common communicator called mpi_comm_world as shown in Fig. 1. The MPIRUN utility creates a distinct communicator, denoted as mpirun_com in Fig. 1, for each group of computational nodes when it loads the executable program onto the processors. Using the mpirun_com communicator, any processor can communicate with others within a group. In order to communicate between different discipline modules or different groups, communicators for inter-discipline and inter-zone communications are also defined using the MPIRUN library. They are denoted by solid and dashed lines with arrows, respectively.

Furthermore, the MPI library has the functionality to create a new communicator for a subset of the allocated processors. Communicators for each discipline are defined so that collective operations can be accomplished within a discipline module. Once a communicator for each discipline is defined, it is quite convenient to do a collective operation within a discipline, such as computing lift and drag coefficients. The communication design shown in Fig. 1 only explains the coupling of three different computational modules, e.g. fluids, structures, and controls. However, if needed, additional modules can be easily added to the process.

The communication design for a single coupled analysis can be further extended to perform multiple analyses concurrently. Figure 2 shows the extension of the communication design for concurrent multiple analyses. As contrast to a single coupled analysis, several processors are assigned to each group. In this figure, each group has N processors, which is the number of different cases running concurrently. They are locally ranked from zero to N-1 within a group. In the first run, the initialization data within a group is distributed from the leading node of each group through a broadcast call using mpirun_com communicator. This makes it easy to distribute initial input data within a group. Once the initial data distribution is completed, each processor of a group will participate in a different analysis. For example, if N cases with different initial angles of attack are concurrently executed, each processor within a group has the same grid data of a zone but computes solutions !
for the different flow conditions, which in this case is a different angle of attack. Within the flow domain, after solving the flow equations at every time step, each zone needs to exchange zonal boundary data with adjacent zones to advance to the next step. For this purpose, data communication is limited only among computational nodes with the same local rank. In this communication strategy, each node can distinguish itself from other nodes assigned to different cases. Therefore, each node having different local rank can participate in different simulations. For multiple multidisciplinary simulations, the same communication strategy is applied for data exchange among the discipline domains. Further details of the HiMAP process are described in Ref. 11

Portability,Scalability and Performance

Figure 3 shows the scalability and performance of HiMAP. To obtain a performance measurement parallel computations are made for a wing-body-empennage configuration. This configuration consists of a single block H-O grid with 180x173x40 points in the streamwise, spanwise, and body-normal directions, respectively. The grid is split into multiple, equally sized zones cut perpendicular to the streamwise direction, with each zone assigned to a separate processor. Timing functions are utilized to exclude initialization and I/O CPU usage, thus only the solver portion of the code is represented.

Figure 3 (a) shows the scalability of the code for steady fluids computations. The solid line represents the ideal linear speedup. Two levels of parallelism are shown in this figure. The single parameter set shows continued splitting of the volume grid from 9 to 36 zones. The multiple parameter set represents executing multiple 9 zone cases concurrently with various angles of attack.

The CPU speed of HiMAP is shown in Fig. 3 (b) in MFLOPS/processor. This shows that the performance is decreased as the number of grid blocks is increased for a single analysis case. This is mainly due to increase in the ratio of communication and computation time per node. The code ran consistently near 26 MFLOPS/processor, which corresponds approximately 100 micro sec/grid pt/step for a single node case on SP2 that composed of RS6000/590 work stations (POWER-2 multi chip with 66.7 MHz clock rate). Another way to measure it is to say that if ENSAERO utilized the full 140 processors on the IBM SP2, one can expect over 3.6 GFLOPS performance.

For the same wing body configuration static aeroelastic computations were made using NASTRAN based sub-structure data. The summary of results from HIMAP are shown in Fig. 4. The aircraft is modeled using 8 fluid zones and 3 substructures, as shown in figure 4. Only the 4 fluid zones for the top half of the domain are shown; the bottom half is similarly modeled. Also, the structure is modeled in rectangular plates (QUAD4s), but plotted as triangular elements.

Figure 4 shows an aeroelastic solution from HiMAP. The right half of the figure colors the structural grid by substructure. The left half of the figure shows the pressure field on the aircraft surface. Both halves of the figure show the realistically deflected aircraft shape. The Scalability and performance for this case is similar to one shown in Fig. 3. Details are given in Ref. 12.

For both single and multiple parameter cases, using this coarse grain parallelism, the code scales up nicely with the increase in the number of processors. The rate of Scalability of this type of coarse grain parallelism is encouraging.

HiMAP is successfully ported to MPP platforms of Sun and SGI (ref. 13). The optimized flow solver performs at a rate of 120 MFLOPS per node on 256 node Origin 2000 MPP platform that can run HiMAP at about 30 GFLOPS. A summary of large scale applications is given in ref. 14 and High Speed Research Program(HSR) reports. For HSR configurations, coupled computations using 10000K fluid grid points and structures data based on 20K finite elements were computed using HiMAP. Supermodular capability of HiMAP is demonstrated by plugging in USM3D (ref. 15) unstructured grid solver in place of patched structured grid solver and computing aeroelastic responses with minimal effort (ref. 16). In Ref. 16 portability of HiMAP to workstation cluster is also demonstrated.

References

1. NASTRAN User's Manual: NASA SP-222 (08), June 1986.

2. Slotnick, J. P. Kandula, M. and Buning P.G.: Navier-Stokes Simulation of the Space Shuttle Launch Vehicle Flight Transonic Flowfield Using a Large Scale Chimera Grid System. AIAA-94-1860.

3. Guruswamy, G. P: ENSAERO - A Multidisciplinary Program for Fluid/Structural Interaction Studies of Aerospace Vehicles. Computing Systems Engineering, Val. 1 nos. 2-4, 1990, pp 237-256

4. Schuster, D., Vadyak, J. and Atta, E.: Static Aeroelastic Analysis of Fighter Aircraft Using a three-dimensional Navier-Stokes Algorithm. AIAA Paper 90-0435, 1990.

5. Gupta, K.K.: Development of Aeroelastic Capability. Jl of Aircraft, Vol 33, No. 5, Sept-Oct 1996.

6. Holst, T.L., Salas, M.D. and Claus, R.: The NASA Computational Aerosciences Program - Toward Teraflops Computing, AIAA paper 92-0558, Aerosciences Conf Reno, Jan 1992.

7. Message Passing Interface Forum, MPI: A Message-Passing Interface Standard,
University of Tennessee, May 1994.

8. Hatay, F., Byun, C., Gee, K., Jespersen, D., Rizk, Y. and Guruswamy, G. P.: A multi-level parallelization Concept for High-Fidelity multi-block Solvers, Supercomputer 97, San Jose Nov. 1997.

9. Fineberg, S.: MPIRUN: A Loader for Multidisciplinary and Multizonal MPI Applications. NAS News, Vol. 2, No. 6, Nov. - Dec. 1994.

10. Eldred, L. Guruswamy, G.P.: User's Guide fir ENSAERO_FE Parallel Finite Element Solver.: NASA/TM-1999-208781.

11. Guruswamy, G.P, Byun, C., Potsdam, M and Farhangnia, M.: User's Manual for HiMAP -
A Portable Super Modular Multilevel Parallel Multidisciplinary Analysis Process.
NASA TM-1999-209578, Sept 1999.

12. Eldred, L. Byun, C. and Guruswamy, G.P.: Integration of High Fidelity Structural Analysis Into Parallel Multidisciplinary Aircraft Analysis. AIAA 98-2075, SDM Conference Long Beach, April 1998.

13. Byun, C. and Guruswamy, G.P.: A Parallel Multi-Block Moving Grid Method for Aeroelastic Applications on Full Aircraft. AIAA-98-4709, Sept. 1998.

14. Guruswamy, G.P: Impact of Parallel Computing on High Fidelity Based Multidisciplinary Analysis. AIAA-98-4709, Sept. 98.

15. Frink, N. T., Pitzadeh, S. and Parikh, P.: An Unstructured-Grid Software System for Solving Complex Aerodynamic Problems: NASA CP-3291, 1995.

16. Raj, P., Goodwin, S. A., Sankar, L.N., Weed, R.A., Parikh, P. C.and Bhat, M. K.: Development and Validation of a Fast, Accurate and Cost-Effective Aeroservoelastic Method on Advanced Parallel Computing System. LG97ER0106, Lockheed Martin Report, July 1997.

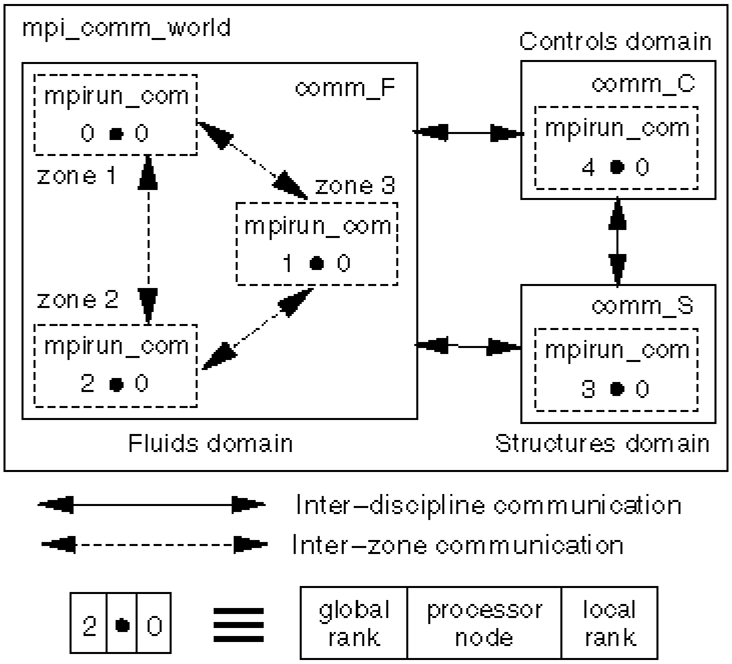


Fig-1

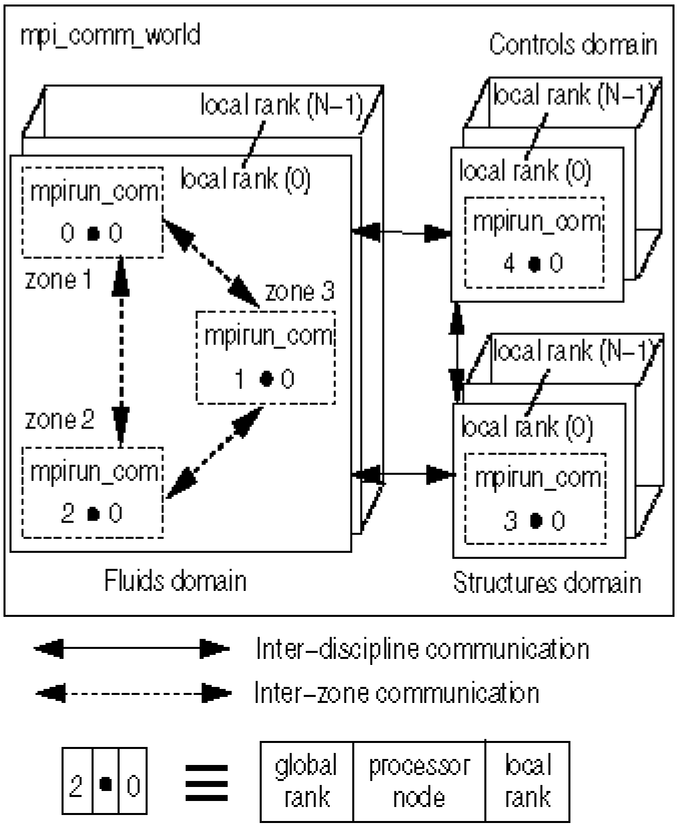
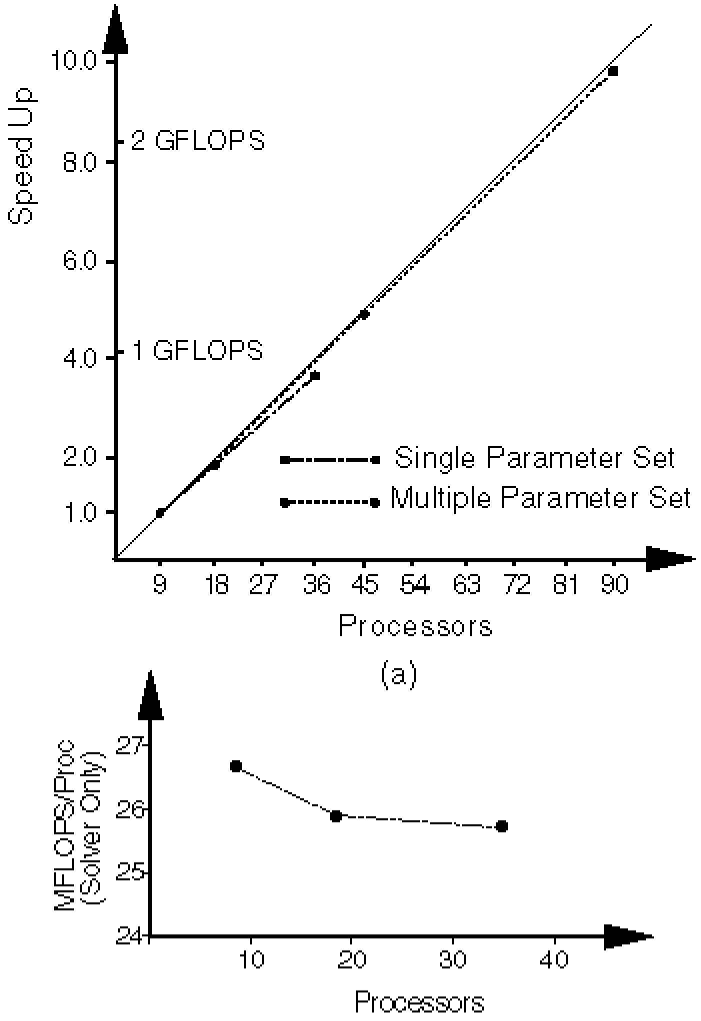

mpi_comm_world
Controls domain
local rank (N−1)
local rank (0)
mpirun_com
0 ● 0
zone 1
zone 3
mpirun_com
1 ● 0
zone 2
mpirun_com
2 ● 0
local rank (N−1)
local rank (0)
mpirun_com
4 ● 0
local rank (N−1)
local rank (0)
mpirun_com
3 ● 0
Fluids domain
Structures domain
Inter−discipline communication
Inter−zone communication
2 ● 0
≡
global rank
processor node
local rank

Fig 2

Speed Up
10.0
8.0
6.0
4.0
2.0
1.0
2 GFLOPS
1 GFLOPS
Single Parameter Set
Multiple Parameter Set
9
18
27
36
45
54
63
72
81
90
Processors
MFLOPS/Proc
(Solver Only)
27
26
25
24
10
20
30
40
Processors

(a)

(b)

Fig 3

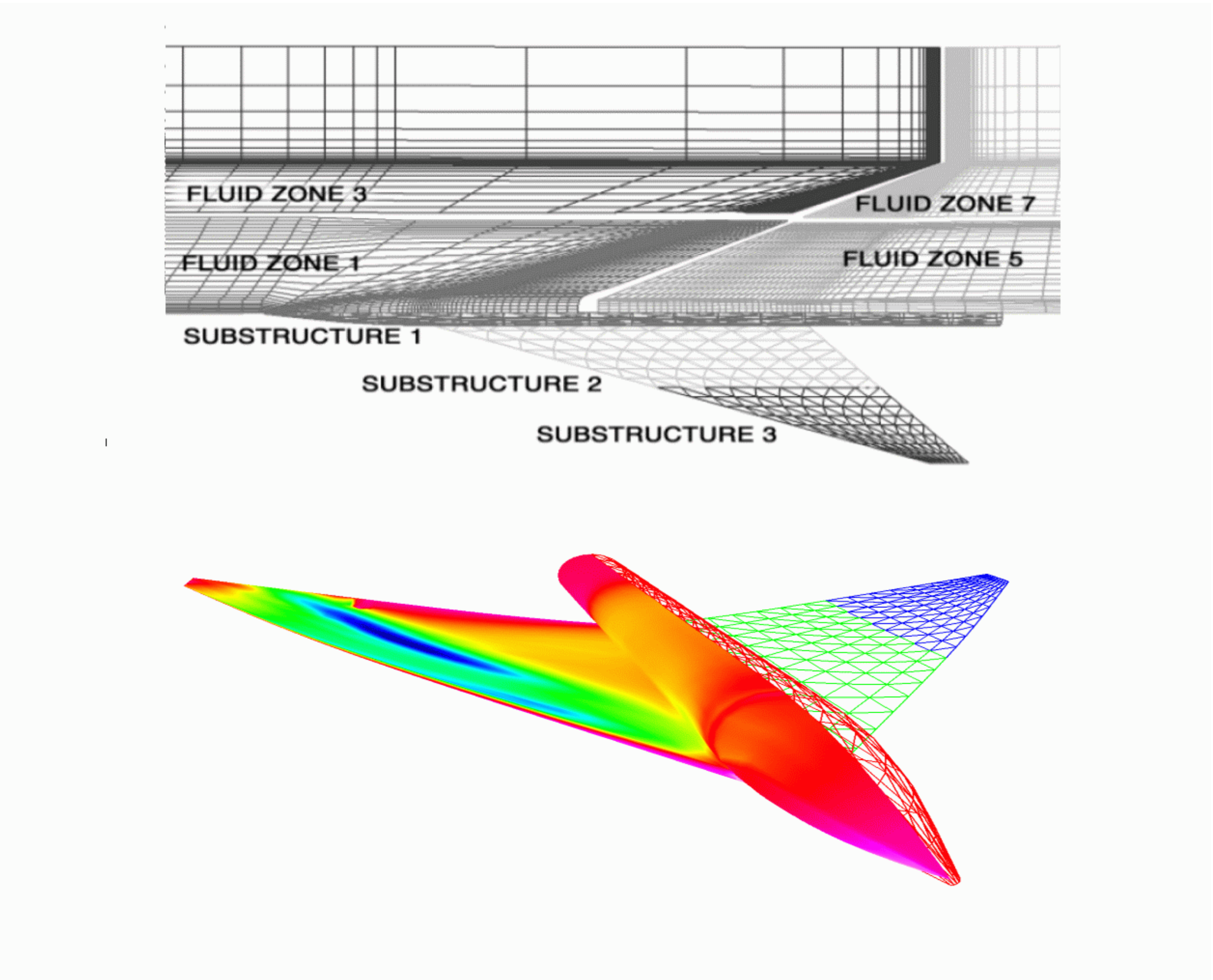

FLUID ZONE 3
FLUID ZONE 7
FLUID ZONE 1
FLUID ZONE 5
SUBSTRUCTURE 1
SUBSTRUCTURE 2
SUBSTRUCTURE 3

Figure 4